Vinum Analytica: Wine Review Insight

Paolo Palumbo

September 13, 2024

Abstract

This report presents a data mining project that analyzes the language used in wine reviews written by sommeliers. We compare the performance of three different models: Decision Tree, Random Forest, and Neural Network. The analysis includes a detailed examination of preprocessing techniques, class contamination removal, and feature vectorization using TF-IDF. We assess the models based on their accuracy and present the results through various evaluations.

1 Introduction

The growing availability of online wine reviews offers a valuable resource for understanding the language used to describe different wine varieties. This study aims to analyze the vocabulary employed in these reviews to classify wines based on the grape variety being reviewed. By examining the textual features of the reviews, we seek to uncover linguistic patterns that correlate with specific grape varieties. Three machine learning models—Decision Tree, Random Forest, and Neural Network—are utilized to explore and classify the relationship between the language used and the wine variety.

2 Corpus Description

The dataset used in this study consists of 130,000 wine reviews, sourced from the Wine Enthusiast website via Kaggle. Each review includes detailed information about the wines, such as the variety, origin, price, and the reviewer's evaluation. The primary focus of this dataset is the textual description of the wines, where reviewers provide sensory insights into the wines' characteristics, such as aroma, flavor, and body.

Not all reviews in the dataset contain complete information, and for the purposes of this study, reviews missing key data, such as the grape variety, were excluded. This cleaned dataset provides a rich source of textual data for analyzing how different grape varieties are described and for training models to classify wines based on these descriptions.

2.1 Exploratory Data Analysis

The dataset contains a diverse range of wine varieties, with some being more common than others.

To ensure a clearer and more coherent representation of wine varieties, a selection was made based on the presence of well-defined grape varieties. Specifically, varieties labeled as 'Blend' were excluded. These varieties represent wines made from blends of different grape varieties rather than a single variety. Their exclusion was motivated by the desire to analyze specific and well-defined grape varieties, avoiding the ambiguity introduced by blends.

Additionally, duplicates were removed from the dataset, and then a minimum representation threshold was applied to the remaining varieties, ensuring that only those with a sufficient number of reviews were included. This process led to a focused selection of well-defined and adequately represented wine varieties, enabling a more accurate and meaningful analysis. By first eliminating duplicates and then excluding underrepresented varieties, we avoided distortions and ensured that the results were based on robust and representative data.

As a result of this selection process, the dataset now includes **27** distinct grape varieties, namely *Barbera*, *Cabernet Franc*, *Cabernet Sauvignon*, *Chardonnay*, *Gamay*, *Gewürztraminer*, *Glera*, *Grenache*, *Grüner Veltliner*, *Malbec*, *Merlot*, *Nebbiolo*, *Petite Sirah*, *Pinot Grigio*, *Pinot Gris*, *Pinot Noir*, *Port*, *Riesling*, *Rosé*, *Sangiovese*, *Sangiovese Grosso*, *Sauvignon Blanc*, *Shiraz*, *Syrah*, *Tempranillo*, *Viognier*, *Zinfandel*. After filtering, the cleaned dataset comprises **75,229** reviews, providing a well-defined and comprehensive basis for further analysis.

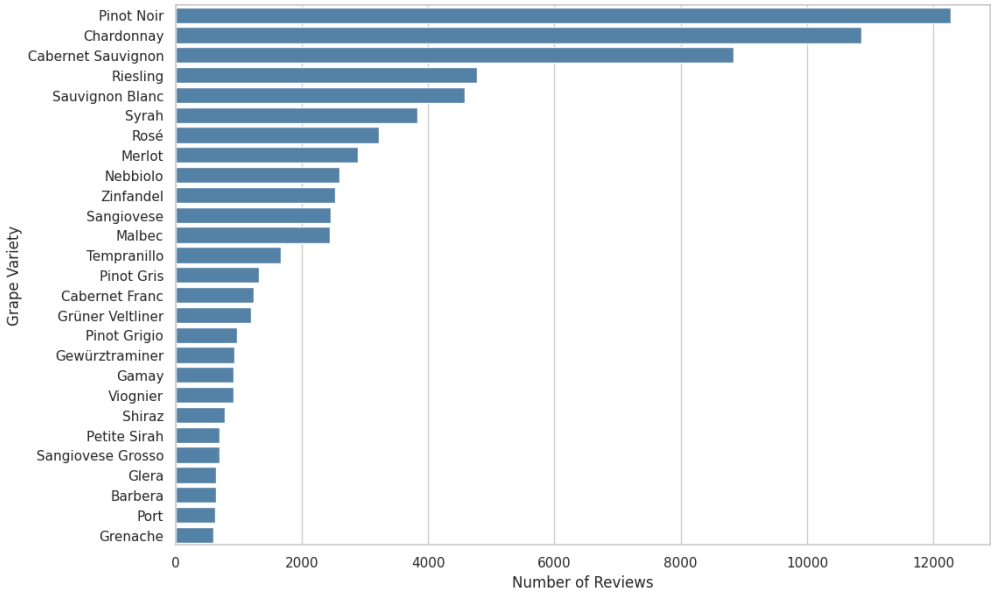

Figure 1: Grapes varieties distribution in the dataset

2.2 Contamination Removal

To further refine the dataset, the keyword for each grape variety was removed from the descriptions. This was done to eliminate any bias introduced by the presence of the variety name in the text. By removing the variety keyword, the analysis can focus solely on the sensory characteristics and descriptions of the wines, without being influenced by the specific grape variety being reviewed.

3 Preprocessing

Data preprocessing is a critical step in any machine learning application. In this project, we applied several preprocessing techniques to prepare the text data for modeling:

- **Text Normalization:** All text was converted to lowercase to ensure uniformity. Additionally, Unicode characters were transformed into ASCII to handle special characters and maintain consistency across the dataset.
- **Tokenization:** Text was split into individual words or tokens.
- **Stop Words Removal:** Common words that do not contribute to the semantic meaning were removed.
- **Stemming/Lemmatization:** Words were reduced to their base or root form.

As shown in Figure 1, the dataset is imbalanced. The most common grape variety is Pinot Noir, and the least common is Grenache.

The dataset was split into a training set and a test set with a 80% - 20% ratio.

3.1 Feature Vectorization

To transform the text data into a numerical format suitable for machine learning, we used the TF-IDF (Term Frequency-Inverse Document Frequency) method. This approach captures the importance of words in the context of the entire dataset and helps in feature extraction.

4 Modeling and Hyperparameter Tuning

Three different categories of models were considered:

- *Decision Trees*
- *Random Forests*
- *Neural Networks*

4.1 Imbalanced Dataset

To overcome the class imbalance in the dataset, a combination of SMOTE oversampling and random undersampling was applied to achieve a balanced dataset. The number of records was limited to

200,000 to control computational time and resources. SMOTE generated synthetic samples for the minority classes by interpolating between existing samples, while random undersampling reduced the number of majority class samples, helping to balance the class distribution without simply duplicating data. This approach improved the model's ability to generalize, particularly for underrepresented classes, while reducing the risk of overfitting. In addition to this balancing strategy, class weights were assigned to further address the imbalance during training, ensuring that minority classes had adequate influence on the model's learning process.

4.2 Model Selection

To identify the optimal hyperparameter settings for each model, we performed a random search by exploring 10 randomly selected hyperparameter combinations for each model (Decision Trees, Random Forests, and Neural Networks). Each combination was evaluated using 6-fold cross-validation on the training set. The hyperparameter search spaces are outlined in Table 1, Table 2, and Table 3.

Accuracy was used as the metric to evaluate the models.

Table 1: Hyperparameter Grid for Decision Tree

Hyperparameter	**Values**
Criterion	gini, log_loss
Min Impurity Decrease	0.0, 1e-8, 1e-10, 1e-12
Max Depth	150, 200, None

Table 2: Hyperparameter Grid for Random Forest

Hyperparameter	**Values**
Number of Estimators	50, 100, 150
Criterion	gini, log_loss
Min Impurity Decrease	0.0, 1e-8, 1e-10, 1e-12
Max Depth	150, 200, None

Table 3: Hyperparameter Grid for Neural Network

Hyperparameter	**Values**
Hidden Size	16, 32, 64
Epochs	6, 8, 10
Learning Rate	0.005, 0.001, 0.0005

5 Experimental Results and Discussion

5.1 Hyperparameter Selection

For each model, we selected the hyperparameter set that achieved the highest average accuracy over the 6-fold cross-validation.

Once the best hyperparameters for each model were identified, we employed the Wilcoxon test to determine whether the differences between the average accuracies of the models were statistically significant. This step ensured that the observed differences were not due to random variation.

Finally, we compared the best results obtained from the three models (Decision Tree, Random Forest, and Neural Network) to assess which model exhibited the highest overall performance.

5.1.1 Decision Tree Results

As shown in Table 4, the Decision Tree models were evaluated using a variety of hyperparameters. The best-performing model, Model 2, used the "gini" criterion with a minimum impurity decrease of 1e-08 and a maximum depth of 200. This combination of hyperparameters resulted in the highest average accuracy of 0.3977.

Other models used different criteria and impurity thresholds, but none matched the performance of Model 2. For example, For example, Model 0 utilized the "gini" criterion with a minimum impurity decrease of 0.0 and a maximum depth of 200, achieving an average accuracy of 0.3954. Model 3, which used the "log_loss" criterion with a minimum impurity decrease of 0.0 and a maximum depth of 200, also performed well with an average accuracy of 0.3969.

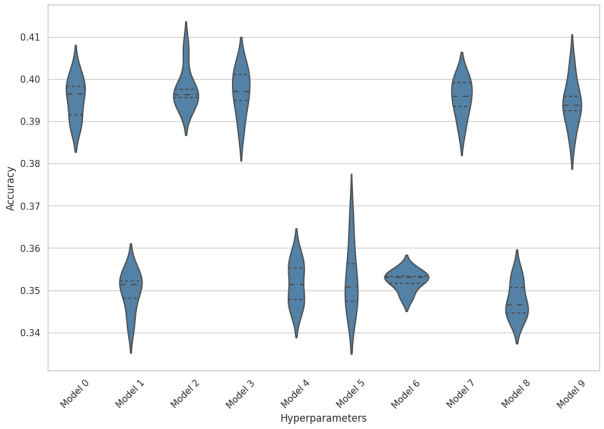

Figure 2: Accuracy distribution for Decision Trees

The Wilcoxon test results indicate that Model 2 shows statistically significant superior performance compared to Models 1, 4, 5, 6, and 8, with p-values below 0.05. However, there are no statistically significant differences between Model 2 and Models 0, 3, 7, and 9, suggesting that their performances are comparable. This confirms that Model 2 is among the best, but not markedly superior to all the other tested models.

5.1.2 Random Forest Results

From Figure 3, it is evident that models 1 and 5 performed better than the others. Specifically, model 5 achieved the highest performance with an average accuracy of 0.5889. The accuracy ranged between 0.5819 and 0.5931, utilizing a configuration of 150 trees (n_estimators), the "gini" criterion, a minimum impurity decrease of 0.0 and a maximum depth of 150. Other models demonstrated lower performance, with model 2 showing the lowest average accuracy of 0.5301.

The Wilcoxon test results indicate that the majority of models (0, 2, 3, 4, 6, 7, 8, 9) exhibit p-values of 0.03125, confirming a statistically significant difference when compared to the best model (index 5). Model 1, however, had a p-value of 0.4375, suggesting no significant difference in performance compared to the best-performing model.

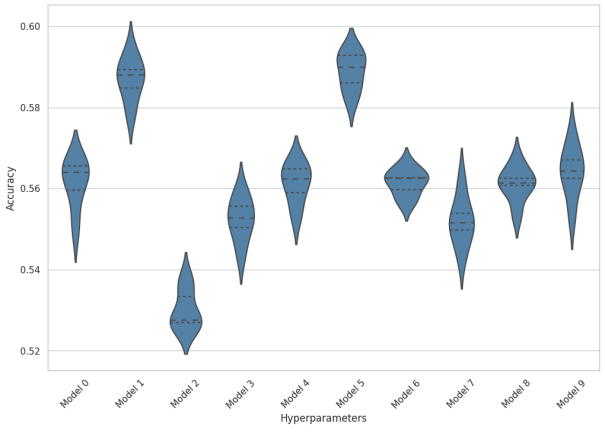

Figure 3: Accuracy distribution for Random Forests

5.1.3 Neural Network Results

For the Neural Network models, the detailed results are illustrated in Figure 4. Model 2 demonstrated the highest performance with an average accuracy of 0.5894. This model was configured with a hidden size of 64, a learning rate of 0.0005, and trained for 8 epochs. The accuracy of Model 2 ranged from 0.5835 to 0.5943.

The Wilcoxon test results indicate that Models 0, 1, 4, 5, and 7 have p-values of 0.03125, indicating statistically significant differences from Model 2. Model 3, with a p-value of 0.15625, shows a less significant difference, and Models 6, 8, and 9, with p-values of 0.0625 and 0.09375, respectively, also demonstrate less pronounced differences. This analysis highlights Model 2 as the top performer, with several other models showing statistically significant, though lesser, performance differences.

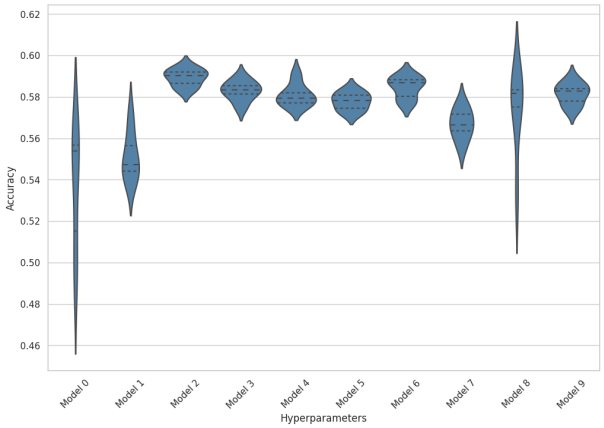

Figure 4: Accuracy distribution for Neural Network

5.2 Best Models Comparison

The performance of the models, as shown in Figure 5, reveals that the Neural Network (NN) achieved the highest accuracy on both validation and test datasets. The NN model reached a validation accuracy of 0.5894 and a test accuracy of 0.5996. The Random Forest (RF) followed, with a validation accuracy of 0.5889 and a test accuracy of 0.5925. The Decision Tree (DT) model had the lowest performance, with a validation accuracy of 0.3977 and a test accuracy of 0.3987.

The Wilcoxon test further confirms these observations. The test result for the DT model against the NN model shows a p-value of 0.03125, indicating a statistically significant difference in performance, with the DT model performing worse. Conversely, the RF model had a p-value of 1.0 when compared to the NN, suggesting no statistically significant difference in performance between RF and NN. This reinforces that while both RF and NN perform comparably, the NN model has a slight advantage. Hence, the NN model is identified as the best-performing model in this comparison.

Index	Criterion	Min Impurity Decrease	Max Depth
0	gini	1e-10	200
1	log_loss	1e-08	200
2	gini	1e-08	200
3	gini	1e-08	150
4	log_loss	1e-10	None
5	log_loss	0.0	150
6	log_loss	0.0	None
7	gini	1e-10	150
8	log_loss	1e-12	200
9	gini	0.0	200

Table 4: Hyperparameters for Decision Tree

Index	Estimators	Criterion	Min Impurity Decrease	Max Depth
0	50	gini	0.0	None
1	150	gini	0.0	200
2	50	log_loss	1e-12	150
3	100	log_loss	0.0	200
4	150	log_loss	0.0	150
5	150	gini	0.0	150
6	150	log_loss	1e-12	150
7	100	log_loss	1e-12	150
8	50	gini	0.0	200
9	50	gini	1e-10	150

Table 5: Hyperparameters for Random Forest

Index	Hidden Size	Epochs	Learning Rate
0	16	8	0.0005
1	16	10	0.001
2	64	8	0.0005
3	32	6	0.0005
4	64	6	0.001
5	64	8	0.001
6	64	10	0.0005
7	64	6	0.005
8	32	6	0.001
9	32	10	0.0005

Table 6: Hyperparameters for Neural Network

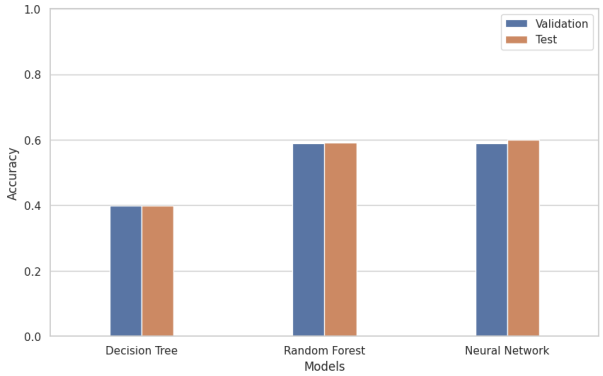

Figure 5: Best Models Comparison

5.3 Discussion of Results

The results of this analysis highlight the superior performance of the Neural Network model compared to both the Decision Tree and Random Forest models. With the highest validation and test accuracies, 0.5894 and 0.5996 respectively, the Neural Network demonstrated its ability to generalize better to unseen data. The confusion matrix in Figure 6 further supports these findings, showing a concentration of correct predictions along the diagonal, indicating strong classification capabilities across the categories.

6 Future Work

6.1 Enhancing the Current Approach

While the current models, particularly the Neural Network and Random Forest, delivered comparable and promising results, there are several areas for improvement. For both models, further fine-tuning of hyperparameters could enhance performance. In the case of the Neural Network, exploring learning rate scheduling, advanced regularization techniques, and different architectures might improve generalization and reduce overfitting. Similarly, for the Random Forest, optimizing parameters such as the number of trees, depth, and split criteria could yield better results.

Additionally, more sophisticated methods for handling class imbalance, such as ADASYN or class-specific loss functions, could improve accuracy for underrepresented classes. Beyond model tuning, exploring alternative approaches for feature extraction from text data, such as word embeddings (e.g., Word2Vec or BERT), topic modeling, or more advanced natural language processing techniques, could provide richer representations of the data and further boost performance across all models.

6.2 Regression Analysis on Wine Prices

In addition to improving the existing classification models, future work could extend the analysis to a regression task focused on predicting wine prices. Implementing regression models to predict continuous variables, such as wine prices, would involve developing and training models specifically tailored for regression tasks, such as Linear Regression, Decision Trees for Regression, and Neural Networks with regression output layers.

The inclusion of price prediction could offer valuable insights into the factors that influence wine pricing, beyond just classification of wine varieties. This would involve preprocessing data for regression, exploring feature engineering techniques, and evaluating model performance using appropriate regression metrics (e.g., Mean Absolute Error, Mean Squared Error). Combining these regression results with the current classification models could provide a more comprehensive understanding of wine reviews and their impact on pricing.

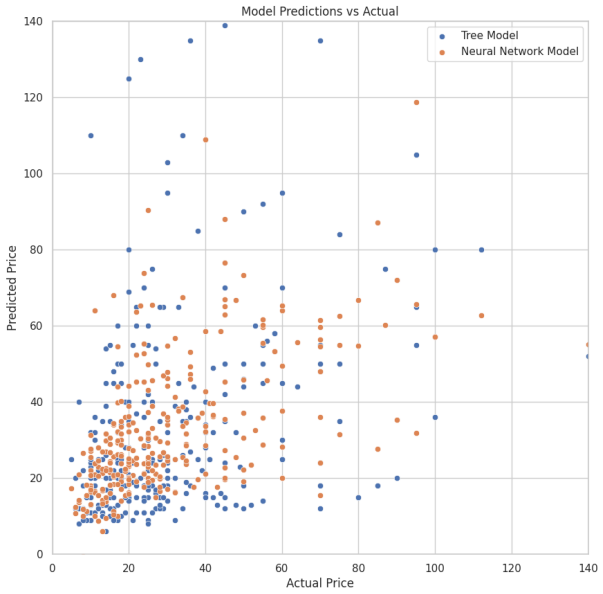

Figure 7: Predicted vs. Actual wine Prices

7 Conclusion

In this study, we investigated various machine learning models for analyzing wine reviews, focusing on

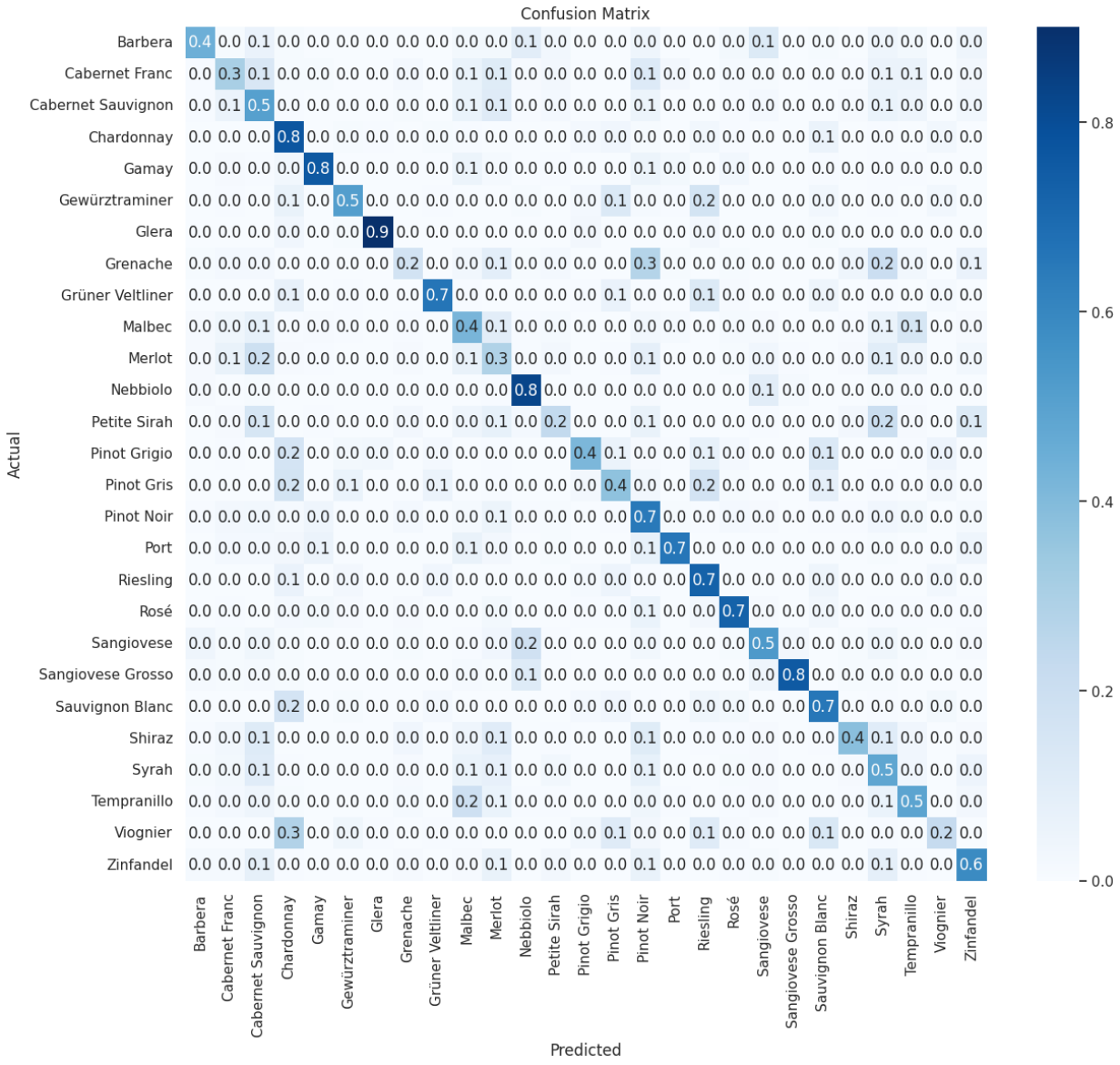

Figure 6: Neural Network Confusion Matrix

classification tasks related to wine varieties. We compared the performance of Decision Trees, Random Forests, and Neural Networks using a rigorous hyperparameter tuning process and cross-validation. Our results indicated that while Decision Trees provided a solid baseline, Random Forests offered improved accuracy and robustness by leveraging ensemble methods. The Neural Networks demonstrated the highest potential for capturing complex relationships within the data, albeit requiring more computational resources and careful tuning to prevent overfitting.

Overall, the findings of this study contribute to a deeper understanding of wine reviews and set the stage for future improvements and expansions in this domain.